Игорь Бессчастный

Лидер платформы API ВТБ

«Особенности открытого Банкинга РФ: эксперименты, болезни роста, первые выводы»

Игорь Бессчастный

Лидер платформы API ВТБ

Отвечаю за реализацию платформ внутренних и внешних API, жизненный цикл API в Банке. Больше 10 лет занимался развитием инновационных сервисов в банках топ-3, включая первые на рынке продукты в области биометрической оплаты и бесконтактных платежей.

Диана Налегач

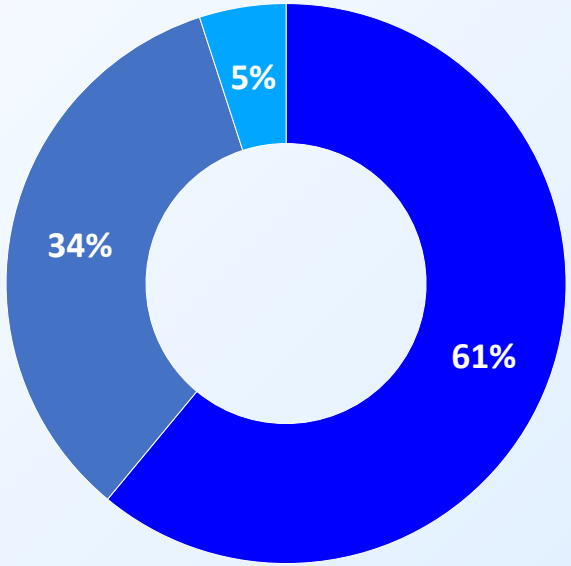
Лидер команды Реестр API ВТБ

Ответственна за разработку и внедрение Реестра API как центрального элемента экосистемы Банка, обеспечивающего реализацию жизненного цикла API на этапе проектирования (Desing Time).

«Реестр API — не просто каталог, а инструмент снижения рисков на уровне проектирования»

Знаете ли вы о концепции открытого банкинга (Open banking)?

- Никогда не слышал(-а) об этом
- Знаю в общих чертах, но мои знания обрывочны
- Точно знаю, что это и для чего



Открытый банкинг, открытые финансы, открытые данные: как они устроены и работают

Открытый банкинг

Банки с помощью открытых API обмениваются между собой данными о состоянии счетов и других финансовых продуктов и сервисов с согласия клиента. У пользователя появляется возможность совершать платежи с разных банковских счетов в едином интерфейсе.

Открытые финансы

В едином интерфейсе пользователю доступны продукты и сервисы не только банков, страховых компаний и брокеров, но и других финтех-игроков.

Открытые данные

К обмену данными присоединятся компании из различных секторов экономики — мобильные операторы, маркетплейсы и др. Также обмен распространяется на государственные данные, медицинские данные и др. (государственные информационные системы и сервисы).

Дорожная карта внедрения открытых API

*данные ДФТ ЦБРФ
Сроки с учетом проработки необходимых нормативных оснований

Дорожная карта внедрения открытых API

Обязательные стандарты Открытых финансов (старт)

Банковский сектор (крупнейшие)

Сведения о счетах и картах физического лица (ФЛ) Сведения о счетах юридического лица (ЮЛ) Сведения об ипотечном договоре ФЛ	Подтверждена востребованность участниками рынка в рамках пилотирования сервисов PFM, BFM, цифровой ипотеки
Инициирование разовых переводов денежных средств (ДС) для ФЛ и ЮЛ Инициирование повторяющихся переводов ДС для ФЛ и ЮЛ	Разработан проект стандарта, подтверждена готовность участников к согласованию и пилотированию в рамках сервисов PFM, BFM в 2025-2026 гг (сроки и участники пилотирования будут уточнены после согласования стандарта – март-апрель 2025)

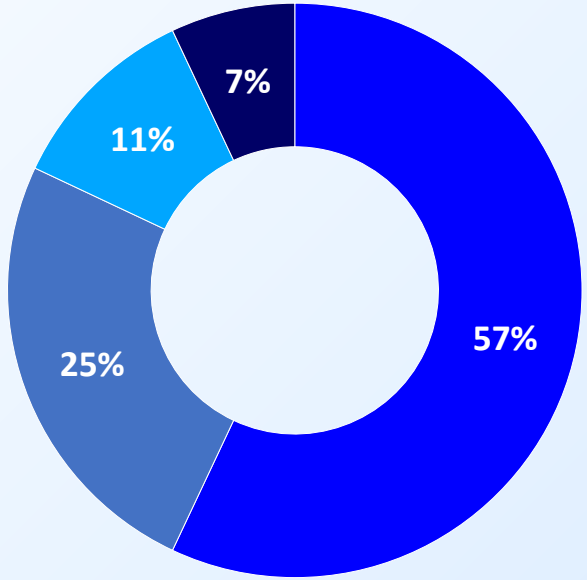
Страховой сектор (крупнейшие)

Сведения о полисе ипотечного страхования ФЛ (жизнь и недвижимость)	Подтверждена востребованность участниками рынка в рамках пилотирования обмена данными при пролонгации договоров страхования и кредитования
Сведения о полисе ОСАГО ФЛ[1] Сведения о полисе КАСКО ФЛ[1]	В рамках опроса[2] граждан больше половины (55%) ФЛ отметили важность получения информации по страховым продуктам в формате одного окна. ОСАГО и КАСКО – самые распространенные виды страхования

*данные ДФТ ЦБРФ

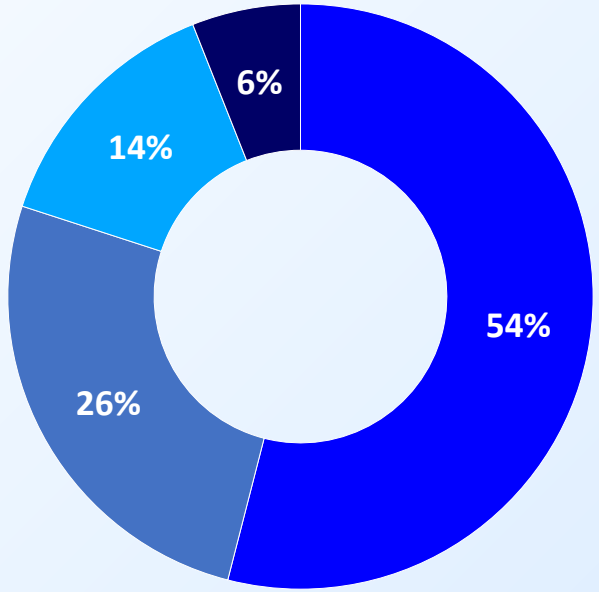
Используете ли вы разные банковские карты для разных целей?

- Я использую карты системно: каждая карта предназначена для определенных операций
- У меня есть несколько карт разных банков, но я не слежу за их разделением по категориям
- Мне достаточно одной карты для всех операций
- Нет, я не разделяю карты по доходам и расходам



Хотели бы вы видеть все счета в разных банках в едином приложении?

- Конечно, это был бы удобный инструмент для управления финансами
- Я бы попробовал(-а), но только если это приведет к экономии бюджета
- Я бы предпочел(-а) оставить все как есть
- Затрудняюсь ответить



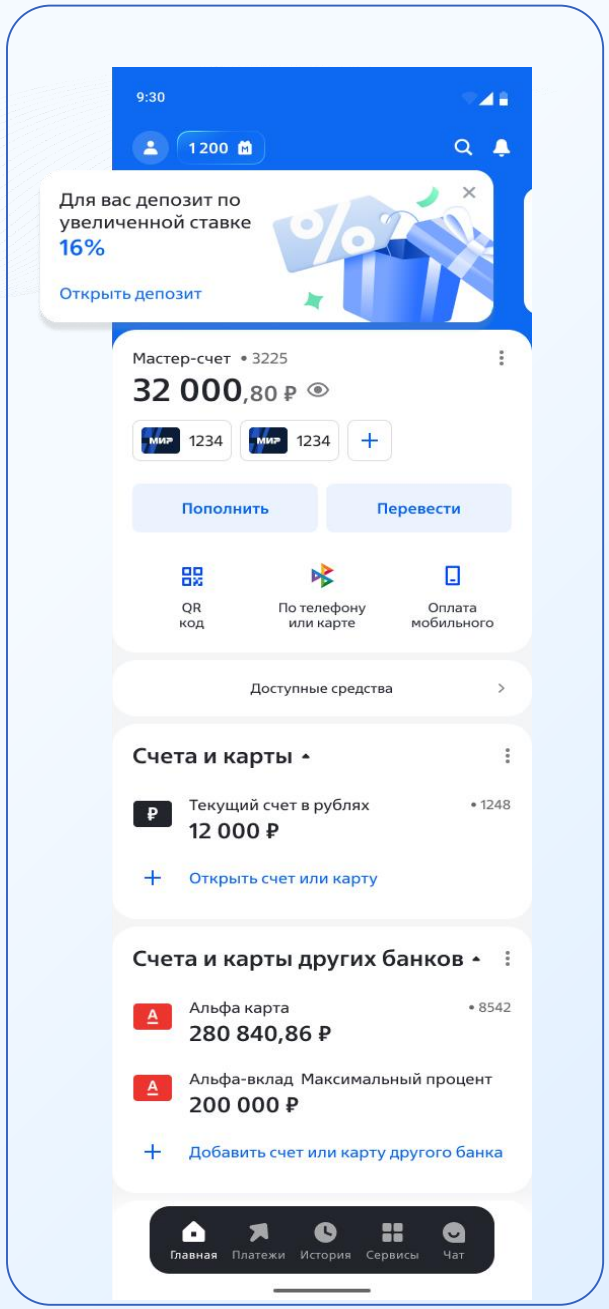
55%

хотели бы консолидировать управление всеми платежами в одном приложении и считают, что такая услуга должна предоставляться бесплатно.

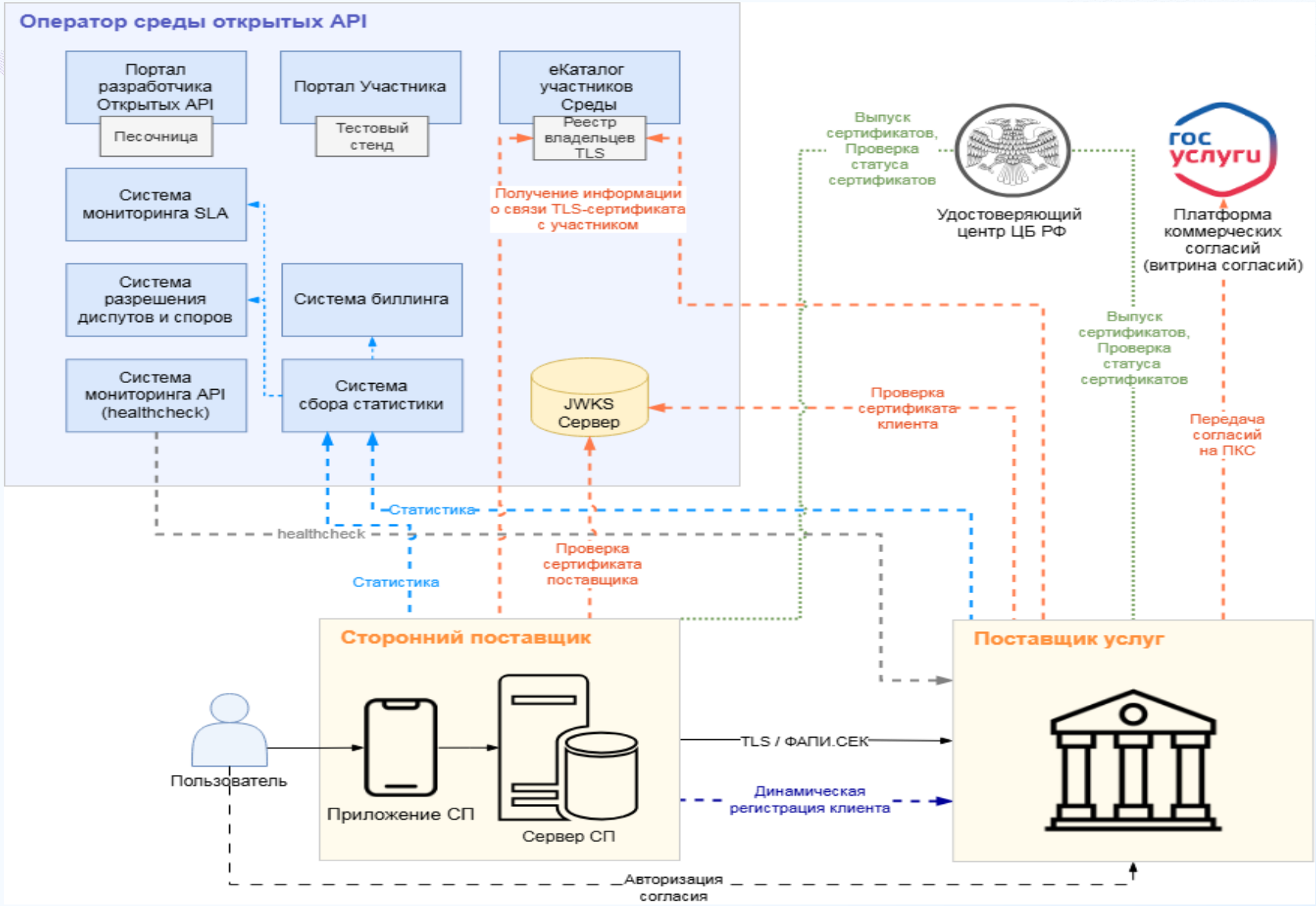
23%

готовы платить за такой функционал ради удобства и экономии времени.

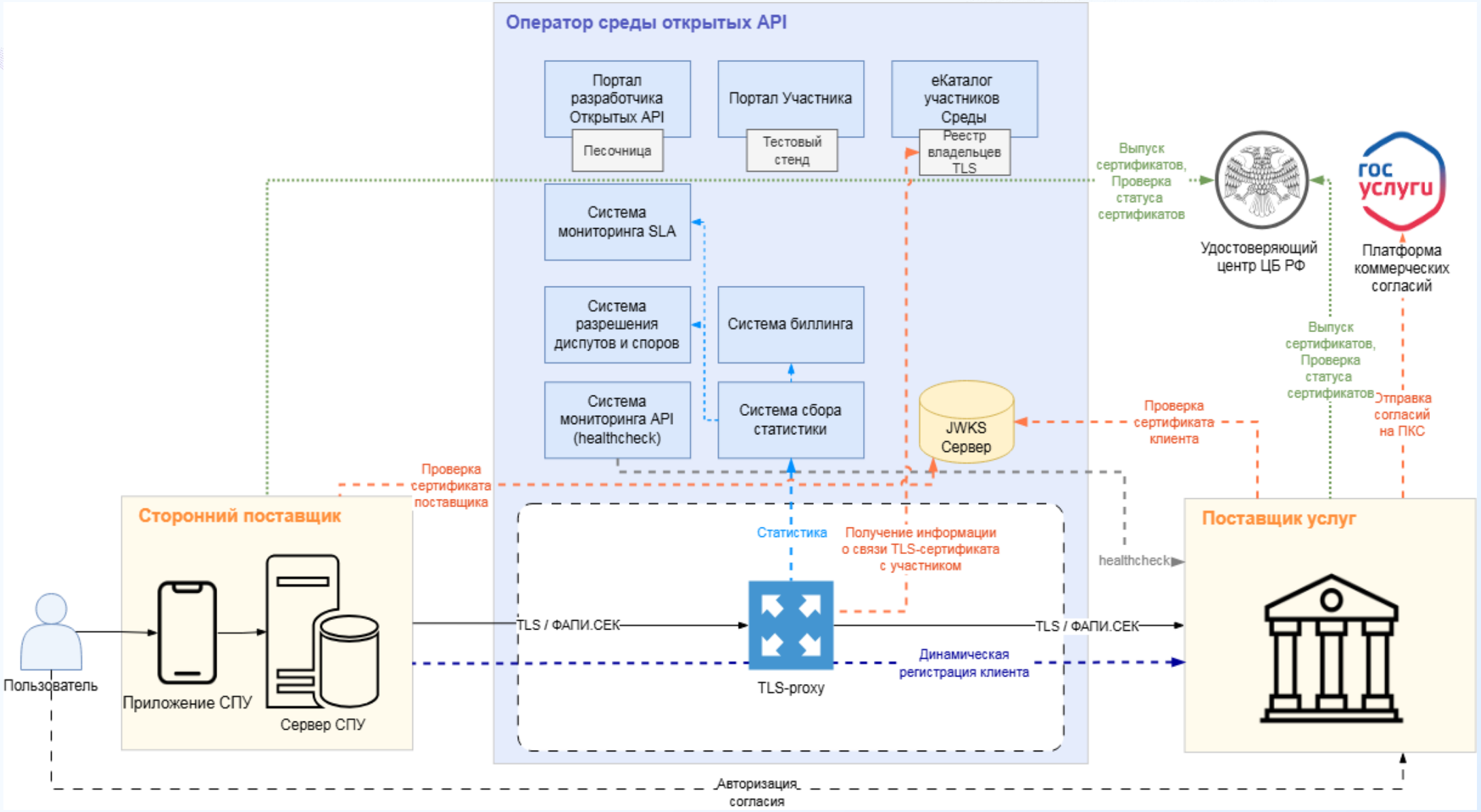
Пример использования открытого банкинга в рамках пилота



Концептуальная техническая архитектура Среды «Full-Mesh+ПКС_v1»



Концептуальная техническая архитектура Среды «+TLS-Proxy»



Вызовы в информационной безопасности в открытых финансовых экосистемах можно классифицировать по нескольким направлениям:

1. Нарушение конфиденциальности – утечки данных, несанкционированный сбор и использование информации третьими сторонами.

Вызовы в информационной безопасности в открытых финансовых экосистемах можно классифицировать по нескольким направлениям:

2. Дискриминация и предвзятость – применение алгоритмов, не учитывающих контекст или содержащих встроенные искажения, что может привести к несправедливым решениям.

Вызовы в информационной безопасности в открытых финансовых экосистемах можно классифицировать по нескольким направлениям:

3. Киберугрозы и мошенничество – недостаточная защищенность данных в финтех-компаниях создает потенциальные уязвимости для пользователей.

Вызовы в информационной безопасности в открытых финансовых экосистемах можно классифицировать по нескольким направлениям:

4. Операционные сбои – технические ошибки и недостаточный контроль процессов могут привести к потере данных, финансовым ошибкам и снижению доступности сервисов.

Вызовы в информационной безопасности в открытых финансовых экосистемах можно классифицировать по нескольким направлениям:

5. Усиление значения ограниченного числа игроков на рынке – доминирование крупных игроков способно ограничить конкуренцию и доступ к инновационным финансовым услугам для пользователей.

Меры противодействия

Усиление защиты API

Внедрение многоуровневых механизмов безопасности, включая контроль доступа, мониторинг аномалий и защиту от DDoS-атак

Обучение пользователей

Проведение регулярных тренингов и информационных кампаний для повышения осведомленности сотрудников и клиентов о методах социальной инженерии и способах защиты от них

Защита от ботов

Использование решений для управления ботами, которые предотвращают автоматизированные атаки и защищают от сбора данных

Мониторинг сторонних API

Регулярная проверка и мониторинг сторонних API и компонентов для обеспечения их безопасности и доступности

Открытый банкинг: безопасные API для всех

Исследование аналитиков ВТБ, подготовленное с привлечением экспертного сообщества

*после перехода по QR-кода необходимо выбрать опцию «Скачать все одним zip-архивом»

Безопасность API в Банках: почему её нельзя игнорировать?

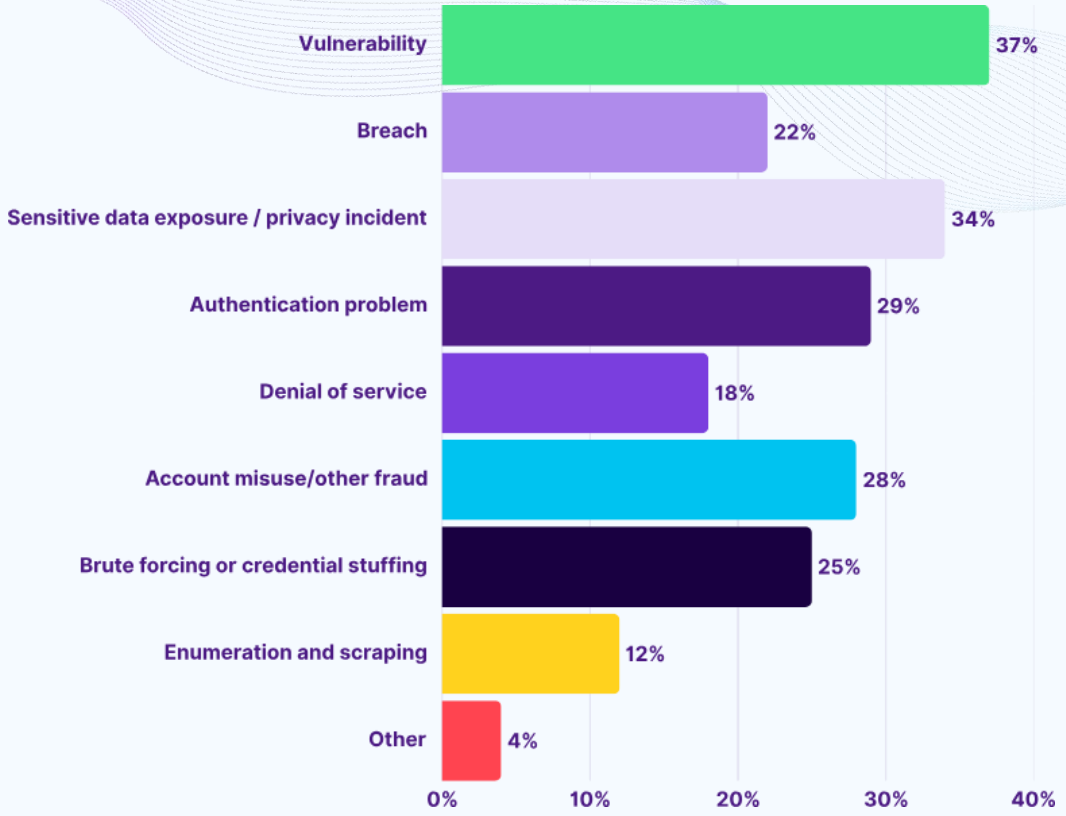


Рис. 1. Какие проблемы безопасности в производственных API вы обнаружили за последние 12 месяцев?

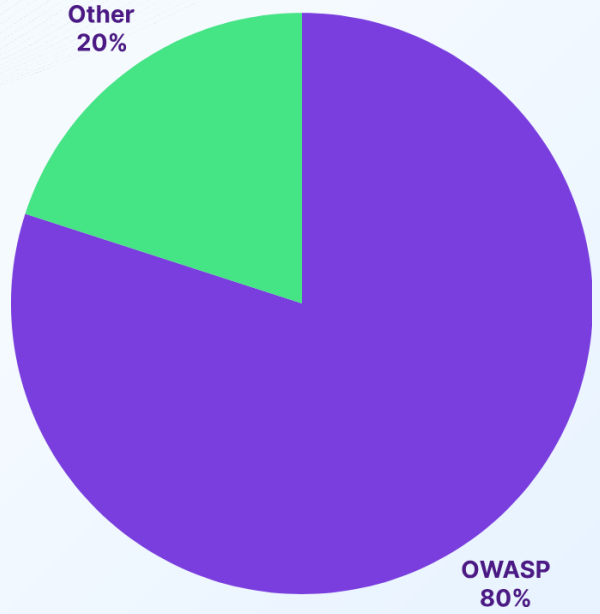


Рис. 2. Данные клиентов SALT: попытки атак с использованием OWASP API Security Top 10 по сравнению с другими типами прикрепления

«Если API спроектирован небезопасно — всё остальное уже не спасёт»

https://salt.security/api-security-trends

Безопасность API в Банках: неутешительные тренды

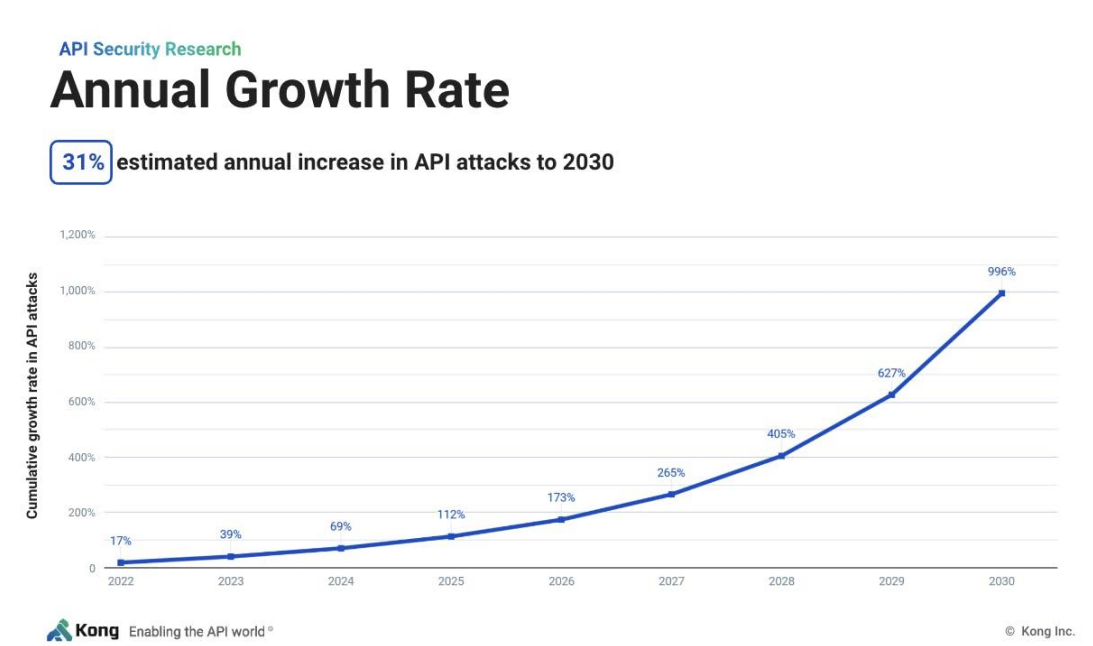


Рис. 1. Прогнозируется всплеск атак на API в течение следующего десятилетия

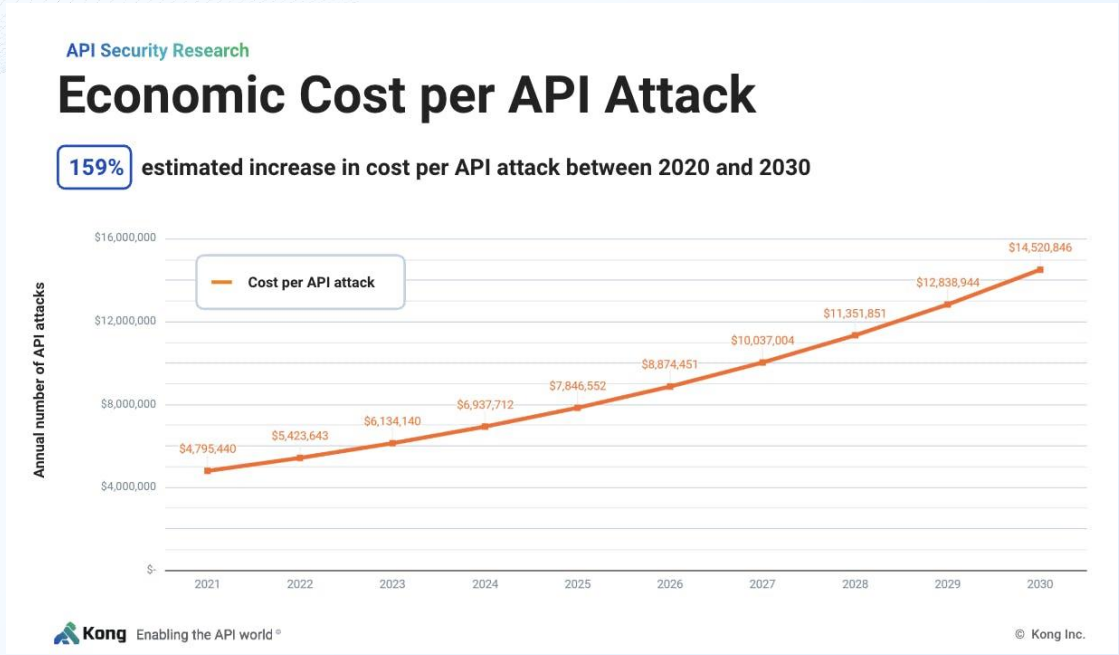


Рис. 2. Стоимость затрат увеличится на 159% к 2030 году

«Сильное развитие API — большие затраты в обеспечении безопасности»

https://konghq.com/blog/enterprise/apis-are-mission-critical

Реестр API — центральное звено в управлении жизненным циклом API

Общая информация по системе:

- Продукт включает в себя лучшие практики инструментов разработки API: стандартизирует структуру API, применяет к API правила валидации, позволяет автоматически исправлять ошибки.
- Продукт позволяет контролировать нагрузку на интеграционное взаимодействие, фиксировать количество Систем, подключающихся к API

- для системных аналитиков
- для владельцев систем
- для разработчиков продуктов
- для архитекторов Систем
- для сотрудников Сопровождения и DevOPS
- для сотрудников информационной безопасности

Бизнес-задачи, решаемые с помощью инструмента:

- Сокращение трудозатрат в проектировании API и повышение качества API ввиду внедрения стандартов проектирования.
- Сокращение времени на вывод API в промышленную эксплуатацию.
- Сокращение трудозатрат в сопровождении API и при переиспользовании API.

*«**Реестр API** — это не просто каталог. Это инструмент обеспечения качества, прозрачности и безопасности всех API вашей организации.»*

Реестр API — основные возможности

ВТБ

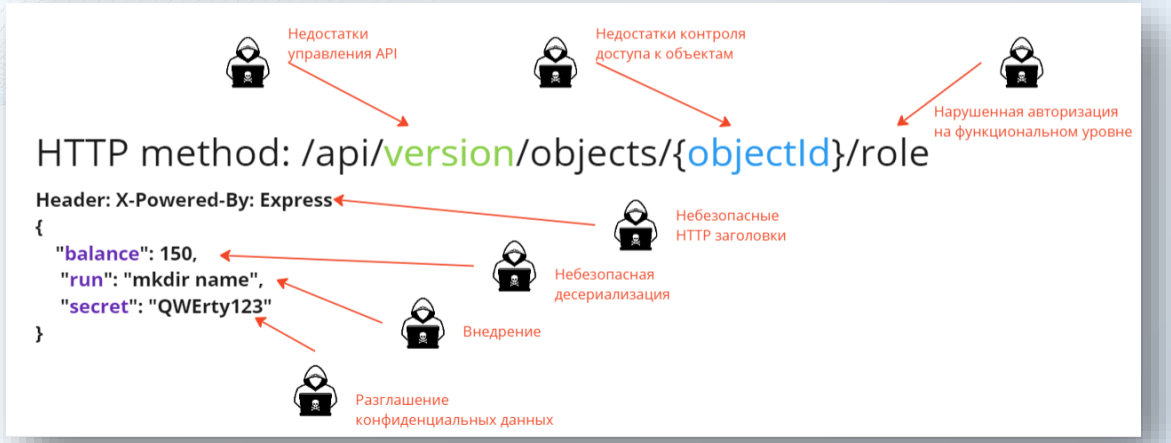
Реестр API

- Code редактор
- Low редактор
- Генерация заглушек
- Добавление плагинов
- Распределение прав доступа
- Валидация API
- НФТ Поставщиков
- Автоисправление ошибок
- НФТ Потребителей
- Версионность API
- Генерация документации
- Подписка на изменения

Риски безопасности на этапе проектирования API

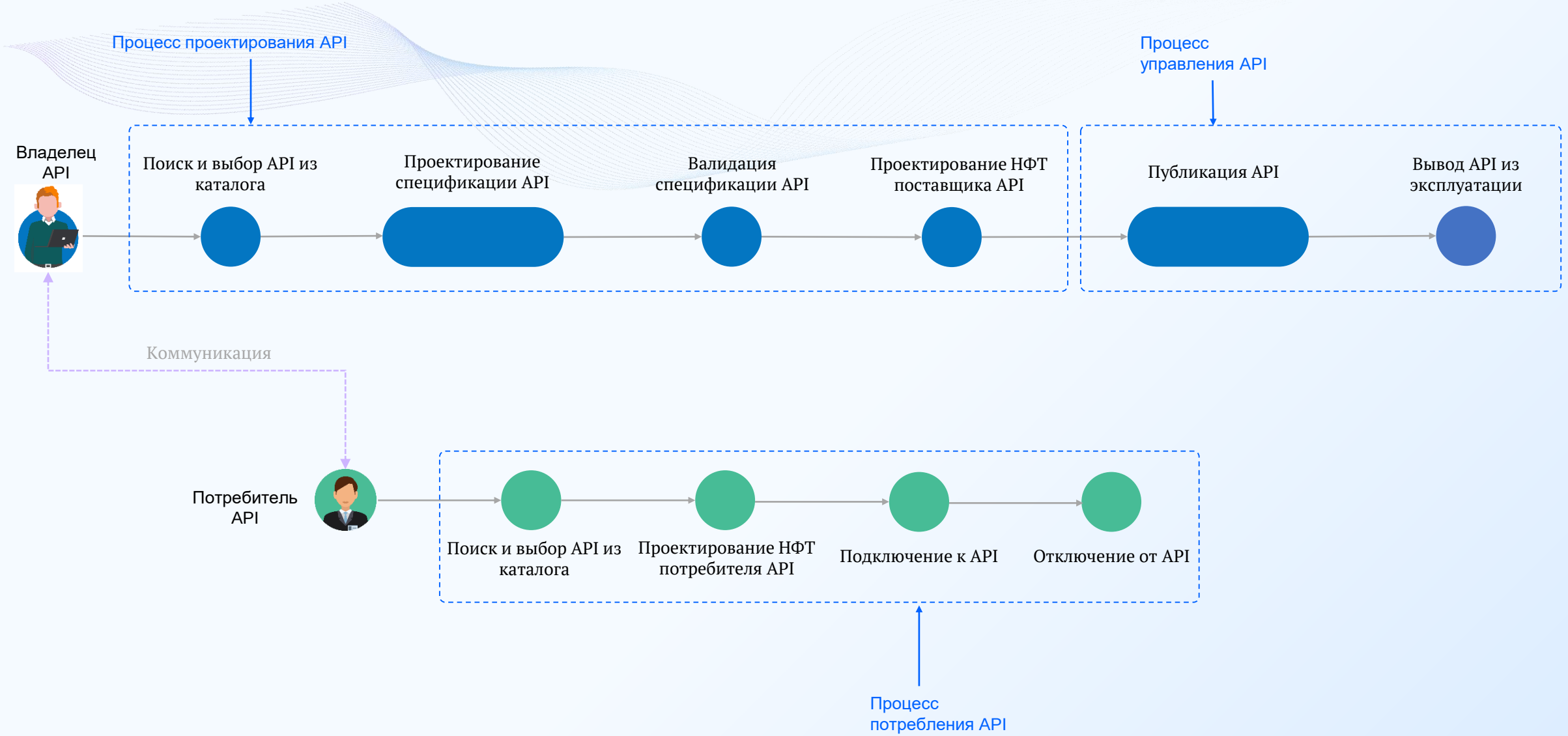
Распространённые ошибки проектирования:

- Отсутствие или некорректная реализация аутентификации и авторизации
- Неправильное определение ролей доступа
- Использование небезопасных методов HTTP
- Передача чувствительных данных в открытом виде
- Неправильная обработка ошибок
- Отсутствие ограничений на размер payload, частоту запросов (rate limiting)
- Неверное описание политик безопасности (scopes, claims, headers)



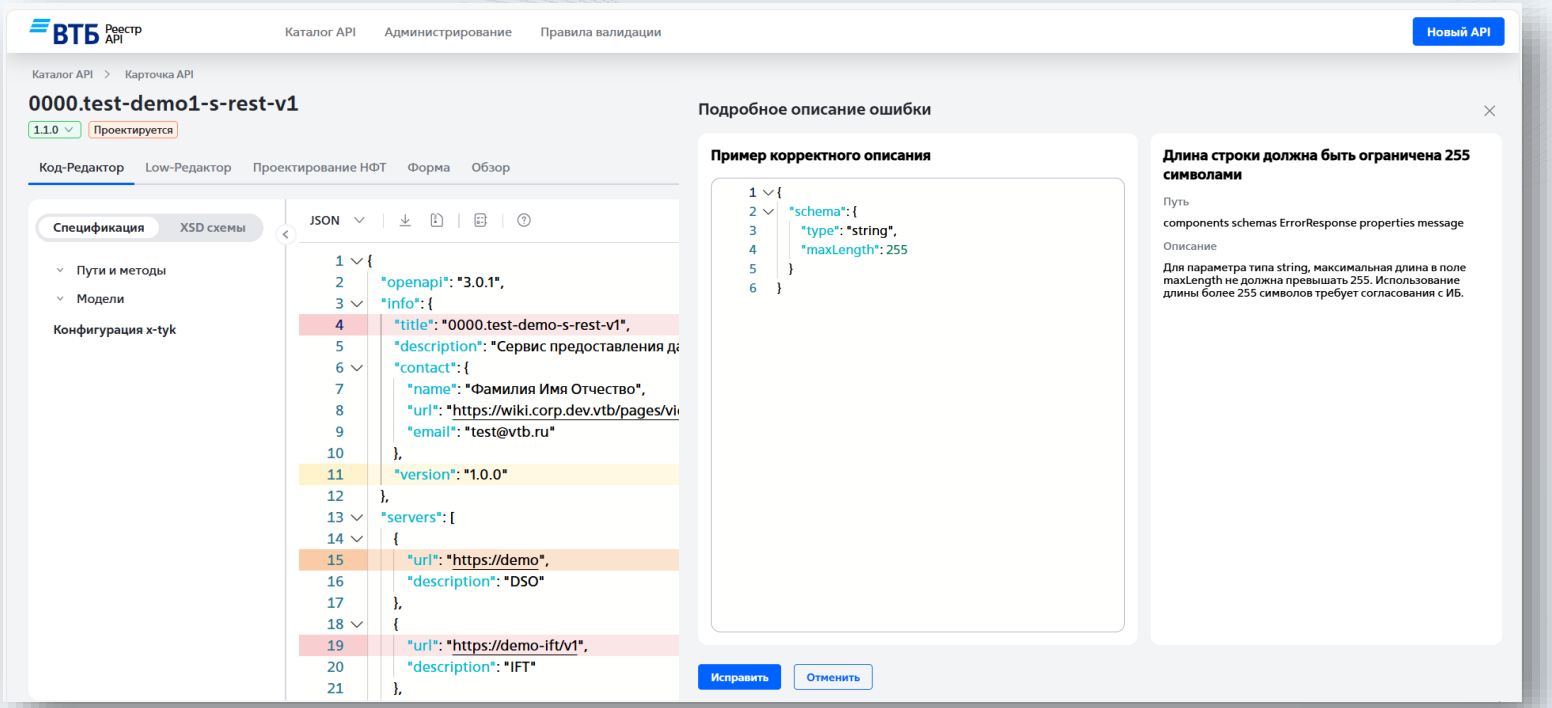
«Ошибка в спецификации — это не просто техническая неточность. Это потенциальная дыра в системе безопасности, которую можно будет исправить только с большими трудозатратами.»

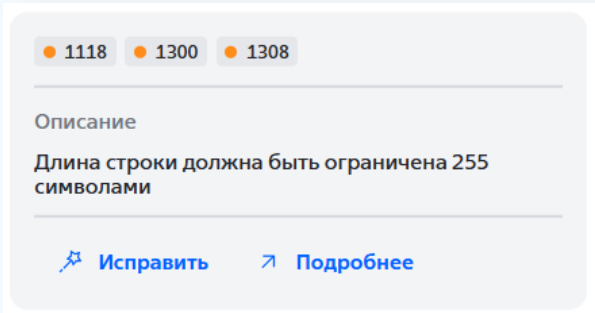
Процесс создания спецификации API в Реестре API

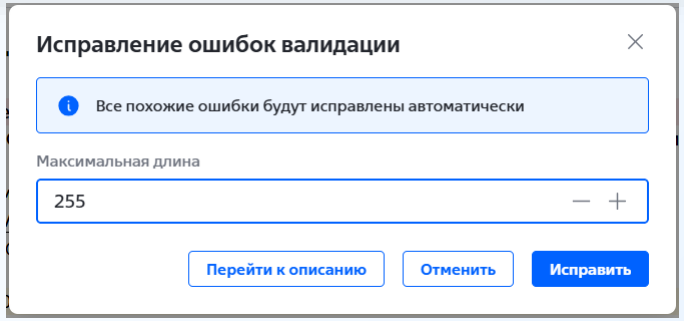


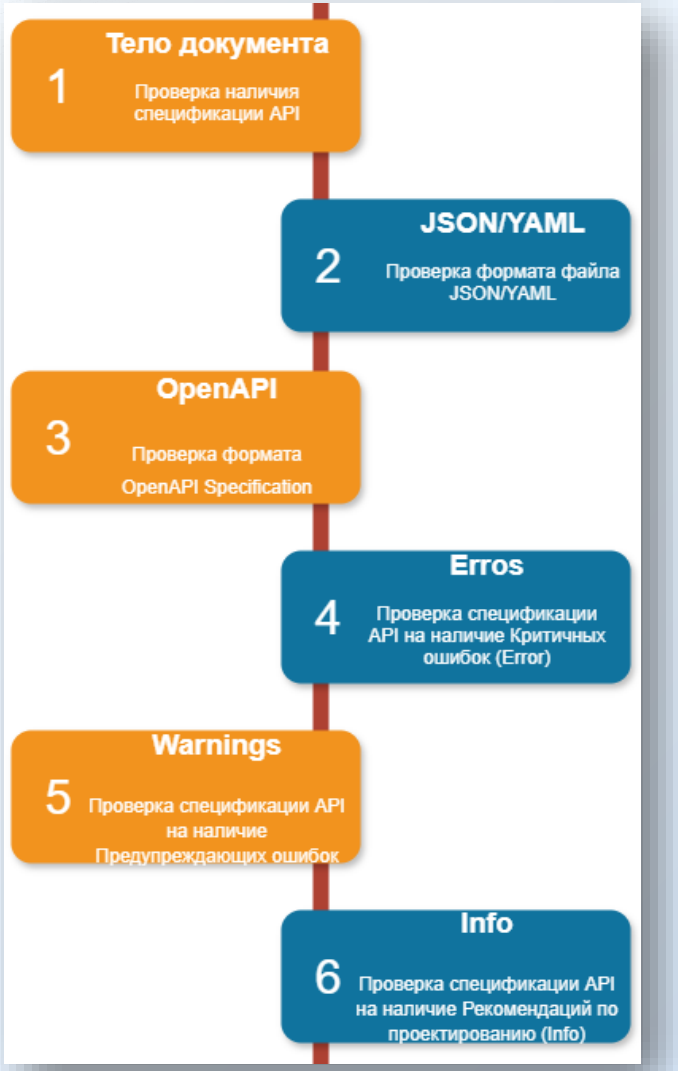
*«**Реестр API** превращает проектирование API из хаотичного процесса в строго регламентированную цепочку, где каждая точка может быть проверена, согласована и протестирована.»*

Как Реестр API помогает находить уязвимости до написания кода?







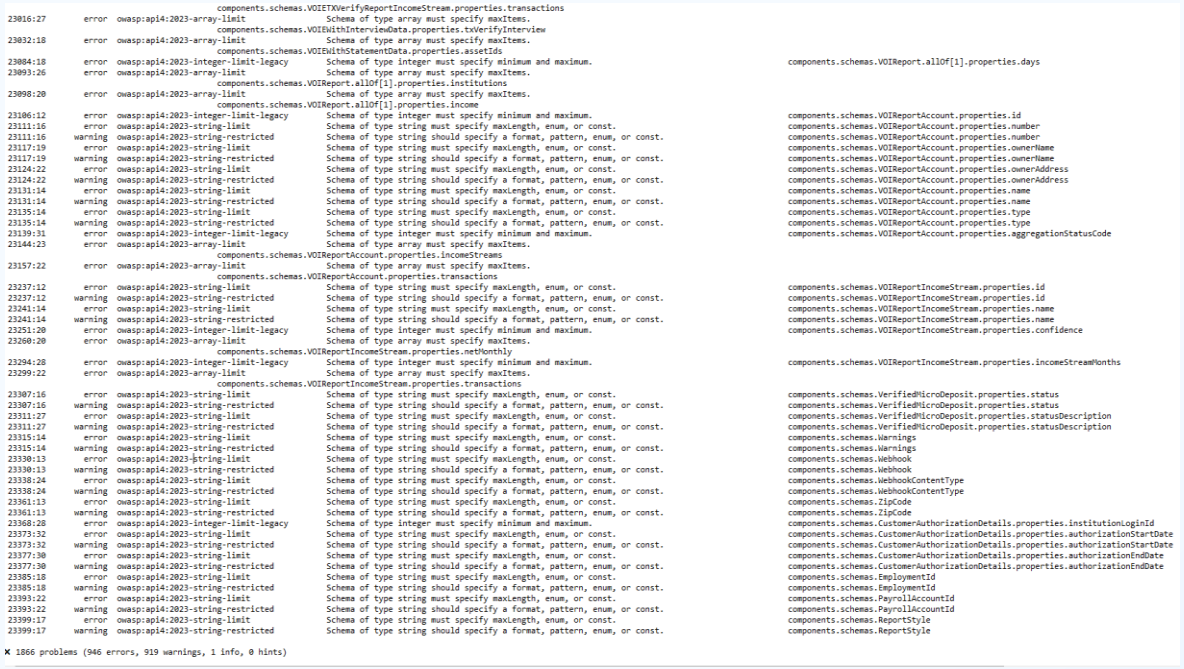


*«**Реестр API** становится первым барьером на пути уязвимостей. Он не даёт им пройти дальше спецификации.»*

Апробируем работу сервиса на примерах

Открытая API проекта Mastercard Open Banking US:

- Наименование API - Open Banking
- 108 вызова API; 93 уникальных ресурсов;
  506 схем запросов и ответов.
- OpenAPI 3.0.3;
- Формат документа YAML;
- Содержит 23 405 строк;
- Подключение производится к хосту https://api.finicity.com;
- Версия спецификации 1.16.1.



```
                                components.schemas.VOIETXVerifyReportIncomeStream.properties.transactions
23016:27      error  owasp:api4:2023-array-limit             Schema of type array must specify maxItems.
                                components.schemas.VOIEWithInterviewData.properties.txVerifyInterview
23032:18      error  owasp:api4:2023-array-limit             Schema of type array must specify maxItems.
                                components.schemas.VOIEWithStatementData.properties.assetIds
23084:18      error  owasp:api4:2023-integer-limit-legacy    Schema of type integer must specify minimum and maximum.                components.schemas.VOIReport.allOf[1].properties.days
23093:26      error  owasp:api4:2023-array-limit             Schema of type array must specify maxItems.
                                components.schemas.VOIReport.allOf[1].properties.institutions
23098:20      error  owasp:api4:2023-array-limit             Schema of type array must specify maxItems.
                                components.schemas.VOIReport.allOf[1].properties.income
23106:12      error  owasp:api4:2023-integer-limit-legacy    Schema of type integer must specify minimum and maximum.                components.schemas.VOIReportAccount.properties.id
23111:16      error  owasp:api4:2023-string-limit            Schema of type string must specify maxLength, enum, or const.           components.schemas.VOIReportAccount.properties.number
23111:16    warning  owasp:api4:2023-string-restricted       Schema of type string should specify a format, pattern, enum, or const. components.schemas.VOIReportAccount.properties.number
23117:19      error  owasp:api4:2023-string-limit            Schema of type string must specify maxLength, enum, or const.           components.schemas.VOIReportAccount.properties.ownerName
23117:19    warning  owasp:api4:2023-string-restricted       Schema of type string should specify a format, pattern, enum, or const. components.schemas.VOIReportAccount.properties.ownerName
23124:22      error  owasp:api4:2023-string-limit            Schema of type string must specify maxLength, enum, or const.           components.schemas.VOIReportAccount.properties.ownerAddress
23124:22    warning  owasp:api4:2023-string-restricted       Schema of type string should specify a format, pattern, enum, or const. components.schemas.VOIReportAccount.properties.ownerAddress
23131:14      error  owasp:api4:2023-string-limit            Schema of type string must specify maxLength, enum, or const.           components.schemas.VOIReportAccount.properties.name
23131:14    warning  owasp:api4:2023-string-restricted       Schema of type string should specify a format, pattern, enum, or const. components.schemas.VOIReportAccount.properties.name
23135:14      error  owasp:api4:2023-string-limit            Schema of type string must specify maxLength, enum, or const.           components.schemas.VOIReportAccount.properties.type
23135:14    warning  owasp:api4:2023-string-restricted       Schema of type string should specify a format, pattern, enum, or const. components.schemas.VOIReportAccount.properties.type
23139:31      error  owasp:api4:2023-integer-limit-legacy    Schema of type integer must specify minimum and maximum.                components.schemas.VOIReportAccount.properties.aggregationStatusCode
23144:23      error  owasp:api4:2023-array-limit             Schema of type array must specify maxItems.
                                components.schemas.VOIReportAccount.properties.incomeStreams
23157:22      error  owasp:api4:2023-array-limit             Schema of type array must specify maxItems.
                                components.schemas.VOIReportAccount.properties.transactions
23237:12      error  owasp:api4:2023-string-limit            Schema of type string must specify maxLength, enum, or const.           components.schemas.VOIReportIncomeStream.properties.id
23237:12    warning  owasp:api4:2023-string-restricted       Schema of type string should specify a format, pattern, enum, or const. components.schemas.VOIReportIncomeStream.properties.id
23241:14      error  owasp:api4:2023-string-limit            Schema of type string must specify maxLength, enum, or const.           components.schemas.VOIReportIncomeStream.properties.name
23241:14    warning  owasp:api4:2023-string-restricted       Schema of type string should specify a format, pattern, enum, or const. components.schemas.VOIReportIncomeStream.properties.name
23251:20      error  owasp:api4:2023-integer-limit-legacy    Schema of type integer must specify minimum and maximum.                components.schemas.VOIReportIncomeStream.properties.confidence
23260:20      error  owasp:api4:2023-array-limit             Schema of type array must specify maxItems.
                                components.schemas.VOIReportIncomeStream.properties.netMonthly
23294:28      error  owasp:api4:2023-integer-limit-legacy    Schema of type integer must specify minimum and maximum.                components.schemas.VOIReportIncomeStream.properties.incomeStreamMonths
23299:22      error  owasp:api4:2023-array-limit             Schema of type array must specify maxItems.
                                components.schemas.VOIReportIncomeStream.properties.transactions
23307:16      error  owasp:api4:2023-string-limit            Schema of type string must specify maxLength, enum, or const.           components.schemas.VerifiedMicroDeposit.properties.status
23307:16    warning  owasp:api4:2023-string-restricted       Schema of type string should specify a format, pattern, enum, or const. components.schemas.VerifiedMicroDeposit.properties.status
23311:27      error  owasp:api4:2023-string-limit            Schema of type string must specify maxLength, enum, or const.           components.schemas.VerifiedMicroDeposit.properties.statusDescription
23311:27    warning  owasp:api4:2023-string-restricted       Schema of type string should specify a format, pattern, enum, or const. components.schemas.VerifiedMicroDeposit.properties.statusDescription
23315:14      error  owasp:api4:2023-string-limit            Schema of type string must specify maxLength, enum, or const.           components.schemas.Warnings
23315:14    warning  owasp:api4:2023-string-restricted       Schema of type string should specify a format, pattern, enum, or const. components.schemas.Warnings
23330:13      error  owasp:api4:2023-string-limit            Schema of type string must specify maxLength, enum, or const.           components.schemas.Webhook
23330:13    warning  owasp:api4:2023-string-restricted       Schema of type string should specify a format, pattern, enum, or const. components.schemas.Webhook
23338:24      error  owasp:api4:2023-string-limit            Schema of type string must specify maxLength, enum, or const.           components.schemas.WebhookContentType
23338:24    warning  owasp:api4:2023-string-restricted       Schema of type string should specify a format, pattern, enum, or const. components.schemas.WebhookContentType
23361:13      error  owasp:api4:2023-string-limit            Schema of type string must specify maxLength, enum, or const.           components.schemas.ZipCode
23361:13    warning  owasp:api4:2023-string-restricted       Schema of type string should specify a format, pattern, enum, or const. components.schemas.ZipCode
23368:28      error  owasp:api4:2023-integer-limit-legacy    Schema of type integer must specify minimum and maximum.                components.schemas.CustomerAuthorizationDetails.properties.institutionLoginId
23373:32      error  owasp:api4:2023-string-limit            Schema of type string must specify maxLength, enum, or const.           components.schemas.CustomerAuthorizationDetails.properties.authorizationStartDate
23373:32    warning  owasp:api4:2023-string-restricted       Schema of type string should specify a format, pattern, enum, or const. components.schemas.CustomerAuthorizationDetails.properties.authorizationStartDate
23377:30      error  owasp:api4:2023-string-limit            Schema of type string must specify maxLength, enum, or const.           components.schemas.CustomerAuthorizationDetails.properties.authorizationEndDate
23377:30    warning  owasp:api4:2023-string-restricted       Schema of type string should specify a format, pattern, enum, or const. components.schemas.CustomerAuthorizationDetails.properties.authorizationEndDate
23385:18      error  owasp:api4:2023-string-limit            Schema of type string must specify maxLength, enum, or const.           components.schemas.EmploymentId
23385:18    warning  owasp:api4:2023-string-restricted       Schema of type string should specify a format, pattern, enum, or const. components.schemas.EmploymentId
23393:22      error  owasp:api4:2023-string-limit            Schema of type string must specify maxLength, enum, or const.           components.schemas.PayrollAccountId
23393:22    warning  owasp:api4:2023-string-restricted       Schema of type string should specify a format, pattern, enum, or const. components.schemas.PayrollAccountId
23399:17      error  owasp:api4:2023-string-limit            Schema of type string must specify maxLength, enum, or const.           components.schemas.ReportStyle
23399:17    warning  owasp:api4:2023-string-restricted       Schema of type string should specify a format, pattern, enum, or const. components.schemas.ReportStyle

✖ 1866 problems (946 errors, 919 warnings, 1 info, 0 hints)
```

№	Угроза безопасности	Мнемокод правила валидации	Количество обнаружений	Класс критичности	Количество сервисов
1	API1:2023 - Broken Object Level Authorization	owasp:api1:2023-no-numeric-ids	4	Error	2
2	API4:2023 - Unrestricted Resource Consumption	owasp:api4:2023-rate-limit-responses-429	201	Warning	98
3		owasp:api4:2023-rate-limit	109	Error	38
4		owasp:api4:2023-string-limit	445	Error	-
5		owasp:api4:2023-string-restricted	495	Error	-
6		owasp:api4:2023-array-limit	148	Error	-
7		owasp:api4:2023-integer-limit-legacy	194	Error	-
8		owasp:api4:2023-integer-format	43	Error	28
9		owasp:api7:2023-concerning-url-parameter	1	Info	1
10	API8:2023 — Security Misconfiguration	owasp:api8:2023-define-error-responses-500	208	Warning	104
11		owasp:api8:2023-define-error-validation	7	Warning	5
12		owasp:api8:2023-define-cors-origin	2	Error	2
13		owasp:api8:2023-define-error-responses-401	8	Warning	7
14	API9:2023 Improper Inventory Management	owasp:api9:2023-inventory-access	1	Error	1

*«**Реестр API** становится первым барьером на пути уязвимостей. Он не даёт им пройти дальше спецификации.»*

Выгода от использования Реестра API с точки зрения безопасности

Безопасность:

- Предотвращение уязвимостей ещё до реализации кода
- Раннее обнаружение ошибок проектирования
- Снижение рисков утечек данных и несанкционированного доступа

Экономия:

- Снижение стоимости исправления ошибок (в десятки раз дешевле на этапе проектирования)
- Уменьшение числа инцидентов в production
- Оптимизация работы команд за счёт автоматизации проверок

Соответствие требованиям:

- Поддержка стандартов: PCI DSS, ISO 27001, OWASP API Security Top 10
- Готовность к аудитам и требованиям ЦБ РФ
- Прозрачность процессов и история изменений

Качество и единообразие:

- Единые стандарты проектирования API по всей организации
- Контроль качества спецификаций
- Обучение разработчиков через автоматические подсказки и правила

*«Использование **Реестра API** — это не просто удобство. Это инвестиция в безопасность, качество и регуляторную зрелость вашей API-инфраструктуры.»*

Заключение: почему безопасность начинается с проектирования?

Основные выводы по обеспечению безопасной разработке API:

1. Угрозы начинаются ещё до написания первой строчки кода

2. Ошибка в спецификации = потенциальная уязвимость в системе

3. Реестр API — первый барьер на пути уязвимостей

4. Валидация безопасности в момент создания спецификации — ключевой элемент защиты

5. API можно сделать безопасным только если начать с этапа проектирования

*«Если хочешь создать безопасный API — начинай с его проектирования. И делай это правильно с помощью **Реестра API**.»*